Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-29. (Canceled).

30. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal, the integral parser comprising:

a word parser that separates a token on the basis of markup and non-markup by referring to a token table for all markup data necessary for a kind of document to be supported, wherein each different token is generated by a corresponding parser, and wherein a same string of the web-document provided to the handheld terminal has a different token according to whether it is a markup or a non-markup in contrast to a general programming language; and a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents model~~on the basis of a document type definition (DTD) of each document, parses each syntax on the basis of the result of parsing the contents model~~, generates a tree-based object on the basis of a graphic user interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal, wherein the word parser separates all of the tokens of a document supplied to the~~integrae~~ integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

31. (Previously Presented) The handheld terminal of claim 30, wherein the word parser comprises:

a comment parser that processes a comment and a space;

a markup start parser that recognizes a markup start tag and generates a token;

an attribute parser that parses an attribute and generates a token; and a parsed character data analyzer that analyzes parsed character data and generates a token, wherein the attribute parser is configured to recognize a name of an attribute or to recognize a value of the attribute.

32. (Previously Presented) The handheld terminal of claim 30, wherein the syntax parser comprises:

an XML verifier that verifies whether a corresponding web-document is composed suitable for each DTD on the basis of the token generated by the word parser; and

a terminal GUI-based object generator that matches the analyzed markup web-document and the GUI of the handheld terminal.

33. (Previously Presented) The handheld terminal of claim 30, wherein the parsing system integrally parses a web-document composed on the basis of any one of SGML and XML related to HTML, XHTML, mHTML, cHTML, WML and HDML.

34. (Canceled).

35. (Currently Amended) A method for parsing a web-document called by a handheld terminal from a web-server to the handheld terminal, the web-document being

composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the method comprising:

outputting information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal;

reading a token from the web-document and parsing the token using the output information;

if the token is not a defined start tag or if the token is a comment or a space as a result of the reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

parsing the attribute of the element from the token, storing Graphic User Interface (GUI)-related information of the element, and parsing contents of the element;

as the result of the parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears;

in the case that the contents of the element are the parsed character data, performing a mapping operation so as to represent a GUI model of a specific markup language by a GUI ~~(Graphic User Interface)~~ of the handheld terminal regardless of the specific markup language, and thereby matching the parsed character data to the GUI of the handheld terminal; and

in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears, terminating, and if the end tag corresponding to the defined start tag does not appear, ignoring and returning, wherein the element supported by the handheld terminal for the web-document is separated from the web-document by at least one of defining a token table on the basis of an element supported by the handheld terminal and making the undefined token an UNKNOWN token, or ignoring the undefined token, wherein a word parser of the handheld terminal separates all of the tokens of a document supplied to an integral parser of the handheld terminal on the basis of markup and non-markup by using the token table and a syntax parser of the handheld terminal ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

36. (Previously Presented) The method of claim 35, wherein the parsing comprises:

if the read token does not include the defined start tag, reading the data continuously until the end tag appears, and if the end tag corresponding to the defined start tag does not appear, ignoring the token; and

reading a new token.

37. (Currently Amended) A recording medium in a handheld terminal that stores a program for parsing a web-document called by the handheld terminal from a web-server to the handheld terminal, the web-document being composed of a predetermined markup language and

the handheld terminal being capable of supporting a markup language, the recording medium being read by a computer, the program comprising the functions of:

outputting information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal;

reading a token from the web-document and parsing the token using the output information;

if the token is not a defined start tag or if the token is a comment or a space as a result of the reading, ignoring the token, and when the defined start tag is read, parsing an attribute of an element from the token;

parsing the attribute of the element from the token, storing Graphic User Interface (GUI)-related information of the element, and parsing contents of the element;

if the contents of the element are parsed character data as a result of the parsing, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, reading data until an end tag appears;

in the case that the contents of the element are the parsed character data, performing a mapping operation so as to represent a GUI model of a specific markup language by a GUI ~~(Graphic User Interface)~~ of the handheld terminal regardless of the specific markup language, and thereby matching the parsed character data to the GUI of the handheld terminal; and

in the case that the contents of the element are not the parsed character data, if the end tag corresponding to the defined start tag appears, terminating, and if the end tag does not appear, ignoring and returning, wherein the element supported by the handheld terminal for the web-document is separated from the web-document by at least one of defining a token table on the basis of an element supported by the handheld terminal and making the undefined token an UNKNOWN token, or ignoring the undefined token, wherein a word parser of the handheld terminal separates all of the tokens of a document supplied to an integral parser of the handheld terminal on the basis of markup and non-markup by using the token table and a syntax parser of the handheld terminal ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

38. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal, the integral parser comprising:

a word parser that extracts and separates tokens representing the web-document supplied regardless of a kind of a markup language used to compose the web-document by referring to a token table, wherein each different token is generated by a corresponding parser, and wherein a same string of the web-document provided to the handheld terminal has a different token according to whether it is a markup or a non-markup in contrast to a general programming language; and

a syntax parser that parses a plurality of syntax ~~for~~ of the web-document on the basis of the tokens ~~extracted~~ and ~~separated by the word parser on the basis of~~ a contents model, generates ~~an~~ a tree-based object on the basis of a ~~GUI~~ ~~(~~Graphic User Interface~~)~~ (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

39. (Previously Presented) The handheld terminal of claim 38, wherein the token table comprises:

tokens defined in an XML document;

keywords defined in DTD for all documents provided to the handheld terminal; and

a list of elements which can be supported by the handheld terminal.

40. (Previously Presented) The handheld terminal of claim 38, wherein the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

an attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

41. (Previously Presented) The handheld terminal of claim 38, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied web-document character by character, recognizes a token of the web-document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

42. (Previously Presented) The handheld terminal of claim 38, wherein the contents model means a hierarchy of elements and an attribute list, and is defined in a DTD for all documents provided to the handheld terminal.

43. (Previously Presented) The handheld terminal of claim 38, wherein the syntax parser comprises:

an XML verifier that verifies whether a web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and the GUI of the handheld terminal.

44. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal, the integral parser comprising:

a token table comprising tokens defined in an XML document, keywords defined in a document type definition (DTD) for documents provided to the handheld terminal, and a list of elements, which can be supported by the handheld terminal;

a word parser that extracts and separates tokens of the web-document supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table, wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute; and

a contents model determined by DTDs for the documents provided to the handheld terminal that includes a hierarchy of elements and an attribute list; and

a syntax parser that parses a plurality of syntax ~~for~~ of the web-document on the basis of the tokens ~~extracted~~ and ~~separated by the word parser on the basis of~~ the contents model, generates ~~an~~ a tree-based object on the basis of a ~~GUI (~~Graphic User Interface~~)~~ (GUI) of the handheld terminal ~~through the parsed syntax~~, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

45. (Previously Presented) The handheld terminal of claim 44, the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

the attribute parser that parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

46. (Previously Presented) The handheld terminal of claim 44, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied web-document character by character, recognizes a token of the web-document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

47. (Previously Presented) The handheld terminal of claim 44, wherein the syntax parser comprises:

an XML verifier that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and the GUI of the handheld terminal.

48. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the

handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal and parses the web-document, wherein the integral parser includes a syntax parser that parses a plurality of syntax of the document on the basis of the tokens and a contents model, generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal;

the memory or hard disc that stores information parsed by the integral parser; and

the application program which uses information extracted from the integral parser, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-document by referring to ~~the~~ a token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

49. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising an antenna, a CPU, a peripheral circuit, a vocoder, a memory, and an audio codec, wherein the memory comprises:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal and parses the web-document on the basis of elements, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-document by referring to ~~the~~ a token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute, and wherein the integral parser includes a syntax parser that parses a plurality of syntax of the document on the basis of the tokens and a contents model, generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by the GUI of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal; and

the application program which uses information extracted from the integral parser, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

50. (Currently Amended) The handheld terminal of claim 48, wherein the integral parser comprises:

~~a~~ the token table ~~comprising~~ which comprises tokens defined in an XML document, keywords defined in a DTD for all documents provided to the handheld terminal, and a list of elements, which can be supported by the handheld terminal;

the word parser which extracts and separates all tokens of the web-document supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-document by referring to the token table; and

~~a~~ the contents model which is defined in the DTD for all documents provided to the handheld terminal and meaning a hierarchy of the elements and an attribute list~~; and~~

~~a syntax parser that parses syntax for the tokens extracted and separated by the word parser on the basis of contents model, and generates an object on the basis of the GUI of the handheld terminal through the parsed syntax~~.

51. (Currently Amended) The handheld terminal of claim 50, wherein the word parser comprises:

a comment parser that recognizes a comment or a space and generates a token;

a markup start parser that recognizes a markup start tag and generates a token;

~~an~~ the attribute parser ~~that~~ which parses an attribute and generates a token; and

a parsed character data analyzer that analyzes parsed character data and generates a token.

52. (Previously Presented) The handheld terminal of claim 50, wherein the word parser comprises a token generator and an XML well-formedness verifier, receives the supplied web-document character by character, recognizes a token of the web-document on the basis of the token table, and extracts the token by using the token generator and the XML well-formedness verifier.

53. (Previously Presented) The handheld terminal of claim 50, wherein the syntax parser comprises:

an XML verifier that verifies whether a supplied web-document is composed suitable for each DTD supplied on the basis of the token extracted and separated by the word parser; and

a GUI-based object generator that matches the parsed syntax and the GUI of the handheld terminal.

54. (Previously Presented) The handheld terminal of claim 48, wherein the application program comprises an object based on the GUI of the handheld terminal.

55. (Currently Amended) A method for parsing a web-document called by a handheld terminal from a web-server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the method comprising:

outputting information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal;

using the output information, reading a token from the web-document by referring to a token table, extracting and separating the token;

if the extracted and separated token is not a defined start tag or if the token is a comment or a space, ignoring the token;

when the extracted and separated token is recognized as the defined start tag, parsing an attribute of an element from the token and storing Graphic User Interface (GUI)-related information of the element;

parsing contents of the element after parsing the attribute of the element;

as the result of the contents parsing, if the contents of the element are parsed character data, storing GUI-related information of the contents, and if the contents of the element are not the parsed character data, determining whether an end tag appears;

in the case that the contents of the element are the parsed character data, performing a mapping operation so as to represent a GUI model of a specific markup language by a GUI ~~(Graphic User Interface)~~ of the handheld terminal regardless of the specific markup language, and thereby matches the parsed character data to the GUI of the handheld terminal;

as the result of the previous step, if the end tag does not appear, repeating the above steps, and if the end tag appears, determining whether the end tag corresponds to the defined start tag; and

as the result of the previous step, if the end tag corresponds to the defined start tag, terminating, and otherwise, ignoring and returning, wherein the element supported by the handheld terminal for the web-document is separated from the web-document by at least one of defining a token table on the basis of an element supported by the terminal and making the undefined token an UNKNOWN, or ignoring the undefined token, wherein a word parser of the handheld terminal separates all of the tokens of a document supplied to an integral parser of the handheld terminal on the basis of markup and non-markup by using the token table and a syntax parser of the handheld terminal ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

56. (Previously Presented) The method of claim 55, wherein the attribute parsing comprises:

if the extracted and separated token does not include a defined start tag, reading the data continuously until the end tag appears, thereby ignoring the token; and

reading a new token.

57. (Currently Amended) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal and parses the web-document on the basis of elements, wherein the integral parser includes a word parser that extracts and separates tokens of the web-document supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-document by referring to ~~the~~ a token table, wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute, and wherein the integral parser further includes a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents model, generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by a GUI ~~(Graphic User Interface)~~ of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-document to the GUI of the handheld terminal;

the memory or hard disc which stores information parsed by the integral parser; and

the application program which uses information extracted from the integral parser, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

58. (Currently Amended) A system for a handheld terminal, comprising:

a content provider configured to provide first type web-documents using a first markup language and second type web-documents using a second markup language different from the first markup language; and

a handheld terminal that receives the first and second type web-documents from the content provider, when the handheld terminal calls a web-document from the content provider to the handheld terminal, the handheld terminal being capable of supporting a markup language, wherein the handheld terminal comprises:

an integral parser configured to output information required for an application program of the handheld terminal from data stored in a memory or hard disc of the handheld terminal and parse both the first type documents and the second type documents on the basis of elements to extract information thereof, wherein the integral parser includes a word parser that extracts and separates tokens of the web-documents supplied to the handheld terminal regardless of a kind of a markup language used to compose the web-documents by referring to ~~the~~ a token table, and wherein the word parser includes an attribute parser configured to recognize at least one of a name of an attribute or a value of the attribute, and wherein the integral parser further includes a syntax parser that parses a plurality of syntax of the web-document on the basis of the tokens and a contents model, generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld terminal, the tree-based object corresponding to the application program, performs a mapping operation so as to represent a GUI model of a specific markup language by a GUI ~~(Graphic User Interface)~~ of the handheld terminal regardless of the specific markup language, and thereby matches the parsed markup web-documents to the GUI of the handheld terminal;

the memory or hard disc which stores information parsed by the integral parser; and

an the application program which receives the information extracted from the integral parser, wherein the word parser separates all of the tokens of a document supplied to the integral parser on the basis of markup and non-markup by using the token table and the syntax parser ignores only a markup portion of the element that is not supported by the handheld terminal, including a tag name (element type) and attributes (attribute list), and browses a non-markup portion, including parsed character data for a user.

59. (Previously Presented) The handheld terminal of claim 51, wherein the attribute parser includes a first attribute parser configured to recognize a name of an attribute and a second attribute parser configured to recognize a value of the attribute.

60. (Previously Presented) The handheld terminal of claim 58, wherein in the case of <p align="center">Hello world!</p>, the terminal that does not support p element ignores markup data between "<" and ">" and browses the parsed character data "Hello world!" for the user.

61. (Previously Presented) The handheld terminal of claim 30, wherein the different tokens are as follows: <html> represents an element type, <p>html</p> represents parsed character data, <!--html--> represents a comment.

62. (New) A handheld terminal that parses a web-document based on elements, when the handheld terminal calls the web-document from a server to the handheld terminal, the web-document being composed of a predetermined markup language and the handheld terminal being capable of supporting a markup language, the handheld terminal comprising:

an integral parser that outputs information required for an application program of the handheld terminal using a token table including tokens, keywords, and a list of elements stored in memory or hard disc of the handheld terminal, the integral parser comprising:

a word parser that parses the document using the token table to generate a plurality of tokens, each token being a basic language element that cannot be further divided grammatically; and

a syntax parser that parses a plurality of syntax of the document on the basis of the tokens and a content model and generates a tree-based object on the basis of a Graphic User Interface (GUI) of the handheld, the tree-based object corresponding to the application program.